Guidelines

Description

Provides information and data to educate software development professionals on the concept, applicability and value of design guidelines. Furthermore, this section collects and makes available a set of Design Guidelines to assist software development professionals (architects, designers, developers, QA, auditors, etc.) in identifying and removing potential classes of vulnerabilities in the software systems they are building, as well as developing more mature and security-knowledge-aware design practices for future software systems.

Overview Articles

Naam	Tijdstip aanmaak versie	Abstract
Guidelines Overview	12/04/06 16:57:16	All systems have vulnerabilities, either in the technology from which they are constructed or in the behaviors of the people who use them.

Most Recently Updated Articles [Ordered by Last Modified Date]

Naam	Tijdstip aanmaak versie	Abstract
Handle All Errors Safely	21/04/06 10:06:31	Unhandled or incorrectly handled exceptions can introduce vulnerability.
Guidelines Overview	12/04/06 16:57:16	All systems have vulnerabilities, either in the technology from which they are constructed or in the behaviors of the people who use them.
Assume that Human Behavior Will Introduce Vulnerabilities into Your System	4/04/06 14:41:08	People introduce vulnerability.
Do Not Perform Arithmetic with Unvalidated Input	4/04/06 14:36:37	Careless modulo arithmetic can introduce vulnerability.
Never Use Unvalidated Input as Part of a Directive to any Internal Component	4/04/06 14:34:41	Using unvalidated input as part of a directive or command to a subsystem can introduce vulnerability.

All Articles [Ordered by Title]

Naam	Tijdstip aanmaak versie	Abstract
Assume that Human Behavior Will Introduce Vulnerabilities into Your System	4/04/06 14:41:08	People introduce vulnerability.
Be Suspicious about Trusting Unauthenticated External Representation of Internal Data Structures	4/04/06 14:31:54	Trusting unauthenticated externalized data structures can introduce vulnerability.
Carefully Study Other Systems Before Incorporating Them into Your System	4/04/06 14:29:22	Indiscriminate delegation of function to other systems can introduce vulnerabilities.
Clear Discarded Storage that Contained Secrets and Do Not Read Uninitialized Storage	4/04/06 14:25:36	Failing to initialize storage can introduce vulnerability.
Design Configuration Subsystems Correctly and Distribute Safe Default Configurations	4/04/06 14:24:09	Poorly designed configuration subsystems and poor default configurations may produce system vulnerabilities.
Do Not Perform Arithmetic with Unvalidated Input	4/04/06 14:36:37	Careless modulo arithmetic can introduce vulnerability.
Do Not Use the "%n" Format String Specifier	4/04/06 14:32:58	Careless use of "%n" format strings can introduce vulnerability.
Ensure that Input Is Properly Canonicalized	4/04/06 14:22:34	Failure to canonicalize input can introduce vulnerability. Inadvertently canonicalizing input multiple times can introduce vulnerability.
Ensure that the Bounds of No Memory Region Are Violated	4/04/06 14:18:49	Violation of memory bounds can introduce vulnerability.
Follow the Rules Regarding Concurrency Management	4/04/06 14:23:35	Failure to follow proper concurrency management protocols can produce serious vulnerabilities. Concurrent access to shared resources without using appropriate concurrency management mechanisms produces hard-to-find vulnerabilities. Many "functions" that are necessary to use can introduce

		"time of check/time of use" vulnerabilities.
Guidelines Overview	12/04/06 16:57:16	All systems have vulnerabilities, either in the technology from which they are constructed or in the behaviors of the people who use them.
Handle All Errors Safely	21/04/06 10:06:31	Unhandled or incorrectly handled exceptions can introduce vulnerability.
If Emulation of Another System Is Necessary, Ensure that It Is as Correct and Complete as Possible	4/04/06 14:30:49	Incorrect or incomplete emulation can introduce vulnerability.
Never Use Unvalidated Input as Part of a Directive to any Internal Component	4/04/06 14:34:41	Using unvalidated input as part of a directive or command to a subsystem can introduce vulnerability.
Treat the Entire Inherited Process Context as Unvalidated Input	4/04/06 14:33:59	Inherited process context that is not validated like other inputs can introduce vulnerability.
Use Authentication Mechanisms, Where Appropriate, Correctly	4/04/06 14:16:43	Incorrectly using, or failing to use, authentication mechanisms can introduce vulnerability.
Use Authorization Mechanisms Correctly	4/04/06 14:17:34	Incorrect use of, or failing to use, authorization mechanisms can introduce vulnerability.
Use Well-Known Cryptography Appropriately and Correctly	4/04/06 14:25:11	Failing to use, or inventing your own, cryptography can introduce vulnerability.

Velden

Naam	Waarde
Categories	knowledge